

# 企业级Hadoop平台源码二次开发- (1)

---

## 一、课前准备

---

1. 安装idea
2. 下载hadoop源码  
<https://archive.apache.org/dist/hadoop/common/hadoop-2.7.0/hadoop-2.7.0-src.tar.gz>
3. 将源码导入idea工具（直接导入即可）

## 二、课堂主题

---

本节课给大家讲解Hadoop RPC原理和HDFS启动流程，为后面二次开发打下基础

## 三、课程目标

---

1. 掌握Hadoop RPC原理
2. 掌握HDFS 启动流程
3. 掌握阅读源码技巧（个人）

## 四、知识要点

---

### 1. 项目要点（5分钟）

#### 1.1 项目背景

公司集群已运行一年多，现在集群为满足公司需求，计划将集群扩为300+节点，在过去一年的集群管理中收集到了一些Hadoop集群的bug和性能改造点，故成立了此项目对当前的Hadoop集群进行性能提升和Bug修复。

#### 1.2 项目目标

提升集群性能，并且保证集群4个9稳定。

#### 1.3 学习本项目的意义

- (1) 通过学习Hadoop的源码，掌握分布式系统设计的本质的思想。
- (2) 数据存储平台是大数据里面非常重要的一个环节。
- (3) 架构师的要求

### 2. 项目基础知识（20分钟）

#### 2.1 版本的选择

当前的Hadoop版本已经发展到Hadoop3.x版本了，但是现在业内大的趋势还是用的Hadoop2.X系列，故我们用的也是Hadoop2.X源码(hadoop2.7.0)

## 2.2 阅读大数据源码的思路

1. 掌握其网络通信架构(hadoop RPC)  
分布式, 网络的通信  
Hadoop : Hadoop RPC  
Flink/Spark(1.5) : Akka
2. 场景驱动的方式 (HDFS的源码)
3. 集群启动  
NameNode  
DataNode
4. 写数据的流程 (元数据管理)
5. 读数据的流程
6. 看源码的时候, 要画流程图/边写注释
7. 看核心类注释 (方式之一)
8. 连蒙带猜

## 2.3 Hadoop RPC

RPC (Remote Procedure Call) —远程 (不同进程) 过程 (方法的) 调用。

c/s 客户端/服务端 (不同进程)

客户端调用服务端的方法, 方法的执行 (服务端)

### 代码实现

pom依赖

```
<dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-client</artifactId>
    <version>2.7.0</version>
</dependency>
```

```
/**
 * 协议
 * @author Administrator
 *
 */
public interface ClientProtocol {
    long versionID=1234L;
    void makeDir(String path);
}
```

服务端代码

```
/**
 * RPC服务端
 * @author Administrator
```

```

    *
    */
    public class NameNodeRpcServer implements ClientProtocol {
        /**
         * 创建目录
         */
        @Override
        public void makeDir(String path) {
            System.out.println("服务端: "+path);
        }

        public static void main(String[] args) throws Exception {
            Server server = new RPC.Builder(new Configuration())
                .setBindAddress("localhost")
                .setPort(9999)
                .setProtocol(ClientProtocol.class)
                .setInstance(new NameNodeRpcServer())
                .build();

            //启动服务端
            server.start();

        }
    }
}

```

#### 客户端代码

```

/**
 * RPC客户端
 * @author Administrator
 *
 */
public class DFSClient {
    public static void main(String[] args) throws IOException {
        ClientProtocol namenode = RPC.getProxy(ClientProtocol.class,
            1234L,
            new InetSocketAddress("localhost", 9999),
            new Configuration());

        namenode.makeDir("/user/opt/soft");
    }
}

```

#### 总结Hadoop RPC的特点:

1. RPC指的是不同进程的方法的调用，分别为客户端和服务端，并且是客户端调用服务端的方法，方法的执行是在服务端。
2. 如果实现RPC的调用，必须要有一个协议，协议其实就是一个接口，里面有一个重要的特征，里面得有versionID
3. 服务端必须要实现协议（接口）
4. 凡是JPS命令能看到的服务，都是RPC的服务端

## 5. 如何构建一个RPC的服务端

```
Server server = new RPC.Builder(new Configuration())
    .setBindAddress("localhost")
    .setPort(9999)
    .setProtocol(ClientProtocol.class)
    .setInstance(new NameNodeRpcServer())
    .build();
```

## 6. 客户端是如何获取代理的?

```
ClientProtocol namenode = RPC.getProxy(ClientProtocol.class,
    1234L,
    new InetSocketAddress("localhost", 9999),
    new Configuration());
```

我们猜测一下，NameNode的启动流程里面会有什么样的代码？我们去验证一下！

```
public interface A{
    public long versionID=12L;
    void a();
}
public interface B{
    public long versionID=122L;
    void b();
}

public class NameNodeRpcServer implements A,B{
    public void a(){

    }
    public void b(){

    }
    {
        Server server = new RPC.Builder(new Configuration())
            .setBindAddress("localhost")
            .setPort(9999)
            .setProtocol(ClientProtocol.class)
            .setInstance(new NameNodeRpcServer())
            .build();
    }
}

public class NameNode{
    private NameNodeRpcServer rpcServer;

    public static void main(String[] args){
```

```

rpcServer= new NameNodeRpcServer()
}

}

```

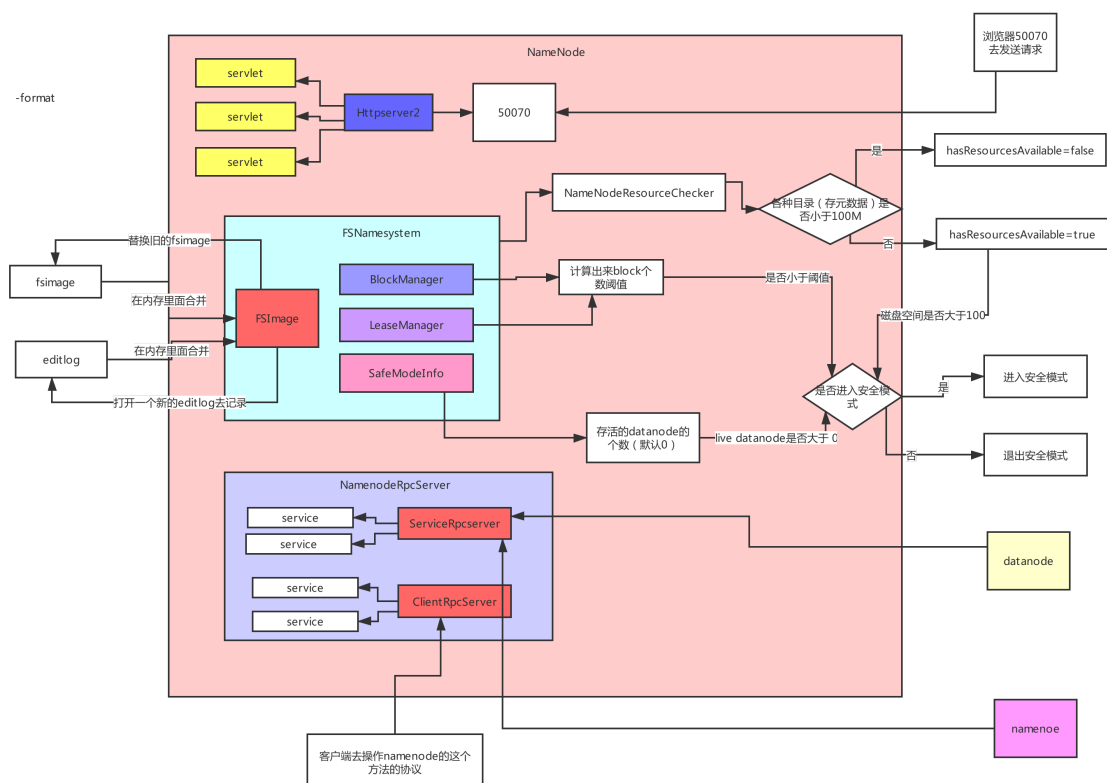
场景驱动：

1. NameNode如何启动的？
2. DataNode是如何启动的？（
  1. 看一下核心的类的注释
  - 2.

## 3. 源码流程讲解

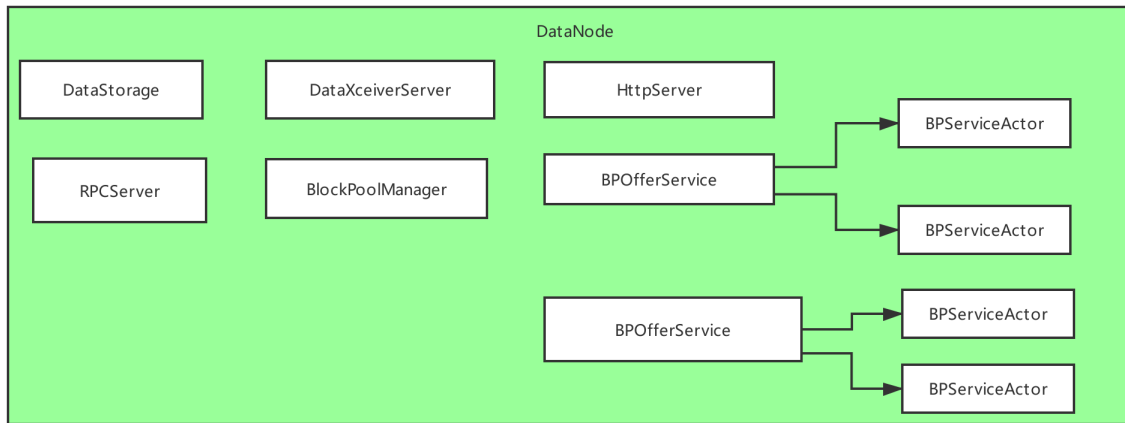
### 3.1 NameNode流程启动剖析（40分钟）

流程图：



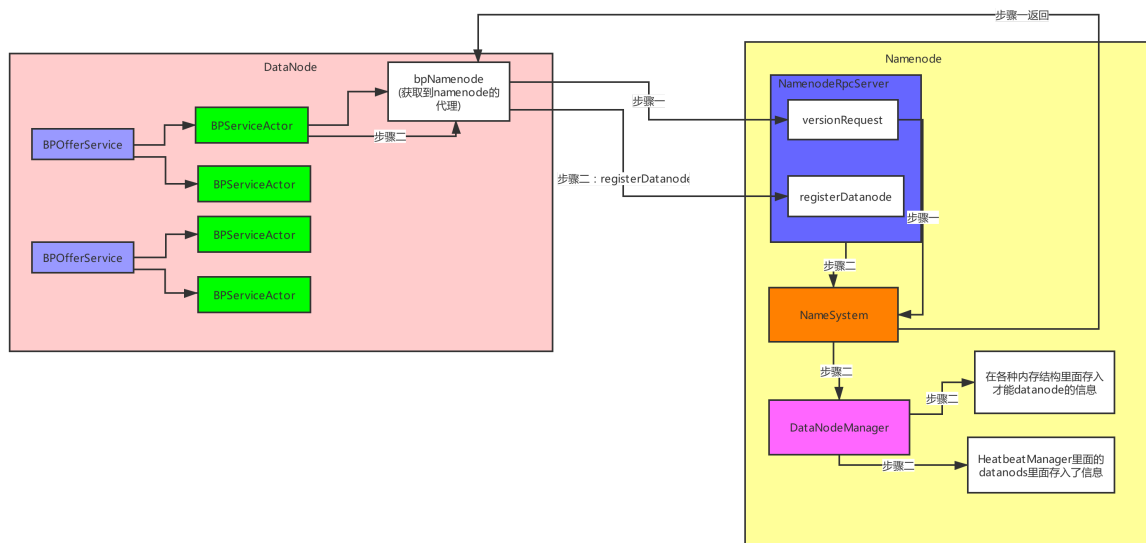
### 3.2 DataNode初始化（10分钟）

流程图：



### 3.3 DataNode注册 (20分钟)

流程图：



## 五、招聘要求 (5分钟)

大数据平台架构师(北京读我科技有限公司) 40000-60000 元/月

分享

有效日期: 2019-09-25

基本要求: 年龄不限 | 性别不限

工作地点: 北京

职位描述:

1、负责大数据平台的设计和开发,负责hadoop、storm、spark、yarn等云计算平台的开发和优化;制定数据架构规范,指导团队落地2、负责数据基础架构和数据处理体系的升级和优化,不断提升系统的稳定性和效率,为公司的业务提供大数据底层平台的支持和保证;3、大数据平台的数据采集、处理、存储以及挖掘分析的架构实现;4、研究未来数据模型和计算框架的创新与落地,包括但不限于以下领域:大规模数据实时化、研发模式敏捷化、数据计算框架轻量化、数据模型组织方式业务化等方面,参与制定并实践团队的技术发展路线;5、建立良好的公司内外的业界技术影响力;参与培养未来数据人才;有效辅导团队,提升数据研发能力。任职资格:1、有很强的数据设计抽象能力,善于从复杂的数据问题中找到关键路径;2、有作为技术负责人系统化解决问题的成功案例;有海量数据实践经验优先;3、熟悉目前正在发展的大数据分布式平台前沿技术的应用;包括但不限于:hadoop、storm、spark、等;4、性格积极乐观,诚信,能自我驱动,有较强的语言表达能力;具备强烈的进取心、求知欲及团队合作精神;具有良好的沟通、团队协作、计划和创新的能力;在数据业界有一定的影响力优先,具有风控经验背景的人优先;5、能够开发创新而实际的分析方法以解决复杂的商业问题

大数据平台架构师-L00426(京东数字科技控... 40000-80000 元/月

分享

职位类型: 技术

发布时间: 2019-07-25

有效日期: 2019-09-25

基本要求: 年龄不限 | 性别不限

工作地点: 北京

职位描述:

职责描述: 1、负责京东商城城市计算平台(数据平台)整体架构设计,大数据技术体系,以及大数据平台的整体规划;2、搭建数据平台技术框架,安排开发人员进行开发,并解决开发过程中细节问题;3、对数据平台安全性,数据质量保障方面进行深入研究,保障平台数据安全和数据质量。任职要求: 1、计算机/应用数学等相关专业、全日制本科以上学历;2、具有6年以上大数据架构设计工作经验,主导过大型企业大数据平台的构建;3、熟悉Hadoop架构与生态圈(如:HDFS、Hive、HBASE、MapReduce、Spark、Flink、Kafka、ElasticSearch、impala等);对hadoop生态圈组件既有广泛的了解,又对某些核心组件有过深入开发经验;4、优秀的沟通理解能力,具备在高压环境下推进工作的能力;5、具备快速学习能力,能快速掌握新的开源技术框架;6、有过时空大数据分析和处理经验优先。



大  
京  
大  
贝  
大  
北  
大  
国  
大



## 职位详情



### 大数据存储研发工程师

60k-90k

北京·海淀区 / 本科及以上 / 5-10年



Nikki 1分钟前来过

BIGO·hrbp

立即沟通

平均1天回复

回复率100%

## 职位描述

存储

岗位职责：

- 1.进行HDFS或者Hbase存储技术栈的源码研究、二次开发，解决实际业务中的问题与挑战；
- 2.打造业界领先的大数据存储系统，为海量数据及其上的大规模数据挖掘、机器学习业务系统提供可靠、高效的支持；
3. 承担千台规模HDFS集群的管理工作，与业务一起解决性能优化、容量规划、预算审计等问题，保障集群高效稳定经济运行。

岗位要求：

- 1.计算机或相关专业本科及以上学历，熟悉Java/Scala/C++/Go等开发语言，2年以上工作经验。
2. 熟悉HDFS或者Hbase源码，有扎实的分布式存储理论基础；
- 3.有HDFS或Hbase社区贡献或者社区活跃者优先；
- 4.有大规模HDFS或者Hbase集群管理和维护者优先；

可以聊

索

投递简历



## 六、总结 (5分钟)

---

1. 阅读源码需要用什么样的一些方式?
2. Hadoop RPC
3. 总结Hadoop RPC
4. 场景驱动的方式 -> namenode -> (类的注释, 验证namenode是否是RPC的服务端)
5. 启动HTTPServer 加载元数据 RPC 资源检查 判断是否进去安全
6. DataNode的启动 -> 场景驱动的方式 -> (类的注释, 验证datanode是否是RPC的服务端)
7. 初始化了很多组件
8. RPC的代码
9. 注册流程

下次课:

10. 心跳

=====

1. 元数据管理 (高可用, 如何抗住亿级流量?)

## 七、作业

---

1. 每位同学至少画一遍流程图

## 八、互动

---